A New OS Architecture for High Performance Communication over ATM Networks
– Zero-copy architecture –

Hiroshi KITAMURA, Kunihiro TANIGUCHI,
Hiromitsu SAKAMOTO, and Takeshi NISHIDA
E-mail: kitamura,taniguti,sakamoto,nishida@nwk.cl.nec.co.jp

C&C Research Labs. NEC Corporation

Abstract A new OS architecture, referred to as *Zero-copy architecture*, for high performance communication is proposed. This architecture dissolved memory copy bottleneck that is a major overhead in protocol processing. This can reduce CPU processing overhead, in addition to realizing high speed data communication. This architecture is shown to be suitable for large volume of data communication like video and image transfer.

1 Introduction

This paper focuses on the communication performance improvement on the end systems ("host" is used hereafter). As many researchers pointed out in their papers [1-6], protocol processing overhead mainly comes from memory copy of both transmission and received data within a host. Although several researchers have tried to overcome this problem [7-9], they have some limitations.

In the paper, we propose a new OS kernel architecture, referred to as "*Zero-copy architecture*," for high performance network communication. This architecture eliminates data copies between applications (user space) and OS kernel (kernel space). Combining "*Zero-copy architecture*" and our experimental ATM NIC (Network Interface Card), which is devised to be able to transfer data directly from and to main memory using DMA, provides high performance end-to-end network communication. Application data can be directly transferred to (from) a network.

This paper discusses some limitations of the other methods for reducing memory copy costs. "*Zero-copy architecture*" and its implementation method are presented. Some experimental results are also shown.

2 Research Background

Jacobson proposes a simple I/O interface (WITLESS)[7], which minimizes both memory copy and control cost between the OS and the network interface. This advanced I/O interface design is adopted by HP's *Afterburner* architecture[8]. The *Afterburner* design aims at avoiding data copy between interface board and main memory. However, data copy between kernel and user spaces is not removed. In addition, since interface and OS kernel share local memory in the *Afterburner* network interface board, a communication application is restrained by this structure. For example, bulk data communication is difficult, because large memory area is needed.

Another interface design for reducing data copy cost is to utilize "*Copy-on-Write*" technique[5], but this technique includes many constrains. Since *Copy-on-Write* technique was designed originally for the same domain memory management, it is difficult to apply this to reduce copy between user and kernel space. Moreover, if an application changes the data, it must be copied onto the other memory area.

3 Design of Zero-copy architecture

"Zero-copy architecture" avoids data copying between kernel and user spaces in main memory at transmission and receiving processes. Essential characteristics of *Zero-copy architecture* are as follows:

> Transmission data in user space is transferred to a network interface driver program without copying. Received data in kernel space into which a network driver program stored is transferred to a user program without copying. In addition, user programs can directly access these memory areas.

Design criteria in "*Zero-copy architecture*" are as follows;

1. Application independent platform: The existing applications, in addition to new multimedia applications, can be easily implemented.
2. Easy migration: The existing major OSs, e.g., UNIX, are easily migrated into the new architecture with minor modifications.
3. Reliability: The new kernel should be reliable in terms of data management.

In order to avoid copy management between kernel and user spaces in communications, memory areas must be shared between the kernel and user processes. The *Zero-copy architecture* realizes this using the memory mapping mechanism.

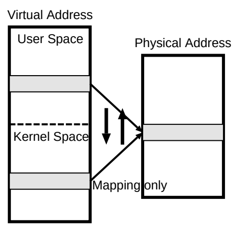

Fig. 1 Zero-copy architecture

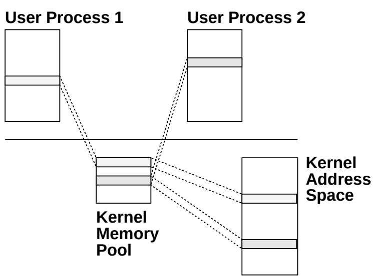

Fig. 2 Memory sharing method

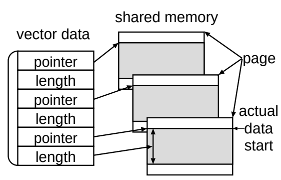

Fig. 3 I/O vector format

As shown in Fig. 1, *Zero-copy architecture* only performs memory mapping in the virtual address space. In *Zero-copy architecture*, virtual addresses of communication data in both the kernel and user spaces are mapped onto the same physical address. The single physical address can be accessed from different virtual addresses by controlling the mapping translation rules of the virtual memory systems. Fig. 2 shows the memory sharing method in the *Zero-copy architecture*. Some areas in the kernel memory pool are exclusively mapped onto multiple user process's user space. A specific area allocated to the kernel space is mapped onto a process's user space temporarily. The control information is exchanged between kernel and application, e.g., pointer to communication data location, in a form of *I/O vector* format shown in Fig. 3. *I/O vector* format is flexible, and it is easy to treat scattered data.

An application interface for *Zero-copy architecture* is as follows. In transmission, an application requests the kernel to allocate and map shared memory area, fill data into the area, and then release the area. On the inbound side, an application requests the kernel to map shared memory area to user space, and release the area after completion of data processing.

4 Implementation and Performance Evaluation

Fig. 4 illustrates our experimental implementation of *Zero-copy architecture* onto the typical UNIX workstation with ATM interface.

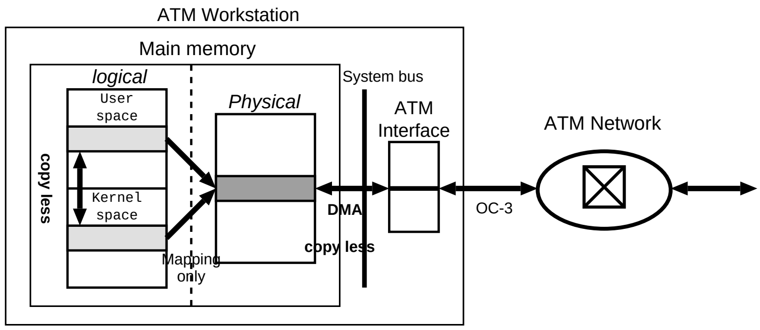

Fig. 4 Zero-copy architecture over ATM LAN

"Mapped device segment" in UNIX is used for data mapping address space. Since the "mapped device segment" has simpler structure and has enough capabilities to implement *Zero-copy segment*. Using "mapped device segment," *Zero-copy architecture* can be realized by a device driver. This leads to enhance portability and extendibility of *Zero-copy kernel*.

The ATM interface hardware can transfer data from and to main memory directly using DMA. That is, there is no local buffer within the network interface.

In order to evaluate the performance of *Zero-copy architecture* itself, it was carried out using a *loop-back interface*. The host processor is RISC CPU with about 130 MIPS power. A user process transmits 3 GB data using a TCP socket. Fig. 5 shows the throughput in changing TCP packet size. Since the TCP checksum computation causes another memory access, we eliminate this in the experiments. The results are **over twice faster** and **220 Mbps throughput**.

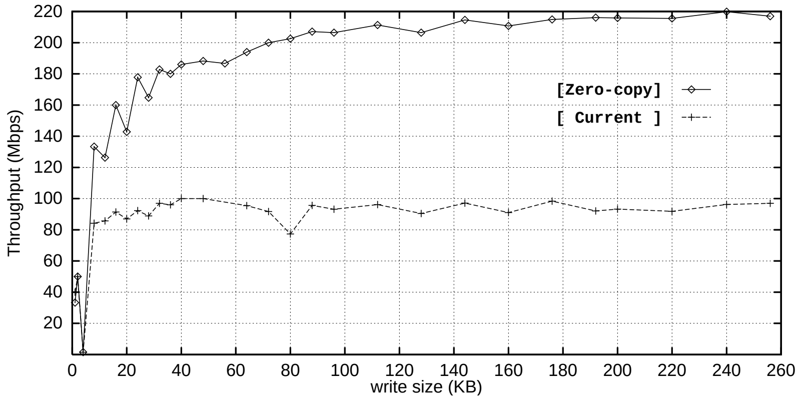

Fig. 5 Zero-copy architecture performance (TCP checksum-off)

5 Conclusion

In this paper, we proposed and explained the design concept and implementation methods of a new OS kernel architecture "*Zero-copy architecture*," to achieve high performance computer communication. This architecture avoids communication data copy on hosts, which is a major bottleneck in a communication process.

References

1. C. Patridge, "Building Gigabit Network Interfaces." connexions 1993.
2. J. K. Ousterhout, "Why Aren't Operating Systems Getting Faster As Fast as Hardware?" Proceedings of 1990 Summer USENIX Conference, Anaheim, California, USA, June 11-15, 1990.
3. J. L. Hennessy, D. A. Patterson, "Computer Architecture: A Quantitative Approach," Morgan Kaufmann, 1990.
4. D. D. Clark, V. Jancobson, J. Romkey, H. Salwen, "An Analysis of TCP Processing Overhead," *IEEE Communications Magazine*, June 1989, pp 23-29.
5. J. M. Smith, G. Q. Maguire Jr., "Measured Response Times for Page-Sized Fetches on a Network," *ACM SIGARCH Computer Architecture News*, Volume 17, No. 5, September 1989, pp 71-77.
6. P. Druschel, M. B. Abbott, M. A. Pagles, L. L. Peterson, "Network Subsystem Design," *IEEE Network Magazine*, Volume 7, No.4.
7. V. Jacobson, "Tutorial Notes from SIGCOMM '90," Philadelphia, USA, September 1990.
8. G. Watson, D. Banks, C. Calamvokis, C. Dalton, A. Edwards, J. Lumley, "Afterburner: Architectural support for high performance protocols," *IEEE Network Magazine*, Volume 7, No.4.
9. P. Druschel, L. L. Peterson, B. S. Davie, "Experiences with a High-Speed Network Adaptor: A Software Perspective," *Proceedings of ACM SIGCOMM 1994*, London, Sep. 1994, pp 2-13
10. C. B. S. Traw and J. M. Smith, "Hardware/Software Organization of a High-Performance ATM Host Interfase," *IEEE Journal on Selected Areas in Communications*, February 1993, Volume 11, No.2